



# LES ALGORITHMES DE MACHINE LEARNING : SVM && KPCA

***PRÉSENTÉ PAR :***

LEILA ESSEBTARI

***ENCADRÉ PAR :***

PR. ABDELHAK MAHMOUDI

## PLAN :

Introduction

Machine à vecteurs de support

Kernel Principal Components Analysis

Conclusion

# INTRODUCTION

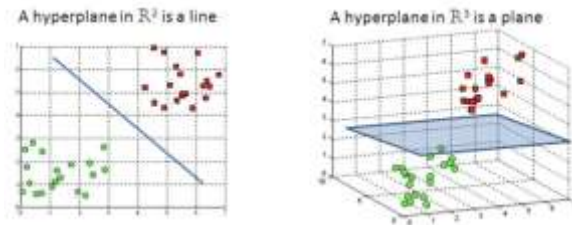
- **L'apprentissage automatique** (en anglais machine learning) ou apprentissage statistique est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, il concerne la conception, l'analyse, l'optimisation, le développement et l'implémentation de telles méthodes.
- Les types d'apprentissage automatique les plus utilisés sont :
  - **L'apprentissage supervisé** : processus consistant à fournir des données d'entrée ainsi que des données de sortie correctes au modèle d'apprentissage automatique., et sur la base de ces données les machines prédisent la sortie.
  - **L'apprentissage non supervisé** : type d'apprentissage automatique dans lequel les modèles sont entraînés à l'aide d'un ensemble de données non étiqueté et sont autorisés à agir sur ces données sans aucune supervision.

# MACHINE À VECTEURS DE SUPPORT : SVM

- Le machine à vecteurs de support , en abrégé SVM, est un algorithme d'apprentissage automatique supervisé utilisé pour les problèmes de classification ainsi que de régression. Cependant, la plupart du temps, il est préféré pour les problèmes de classification.
- SVM a été introduite pour la première fois dans les années 1960 et plus tard improvisée dans les années 1990. Il était très célèbre à l'époque de leur création, et continue d'être la méthode de référence pour un algorithme hautement performant avec un peu de réglage.
- L'objectif de l'algorithme SVM est de créer la meilleure ligne ou limite de décision qui puisse séparer l'espace à  $n$  dimensions en classes afin que nous puissions facilement placer le nouveau point de données dans la bonne catégorie à l'avenir. Cette limite de meilleure décision est appelée **hyperplan**.
- SVM choisit les points extrêmes qui aident à créer l'hyperplan. Ces cas extrêmes sont appelés **vecteurs de support**, et donc l'algorithme est appelé machine de vecteur de support.

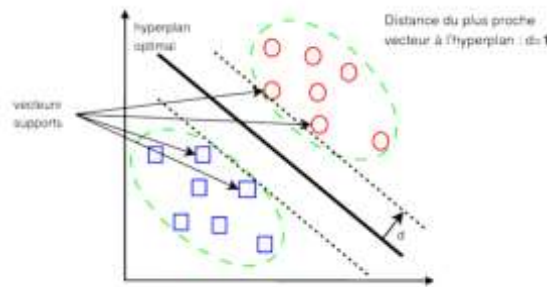
# HYPERPLAN , VECTEURS DE SUPPORT ET MARGE DANS L'ALGORITHME SVM

- **Hyperplan** : les hyperplans sont des limites de décision qui aident à séparer les points de données en différentes classes. Il crée différents côtés, et en fonction du côté où tombe le point de données nouvellement ajouté, il peut être attribué à cette classe particulière. Selon le nombre d'entités, l'hyperplan peut avoir plusieurs dimensions.



**Figure 1** : Hyperplan dans les entités 2D et 3D

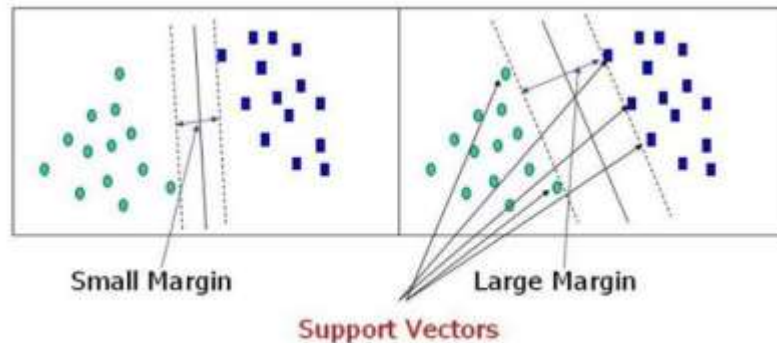
- **Vecteurs de support** : ce sont les points de données extrêmes qui peuvent avoir un impact direct sur la position d'un hyperplan. Ce sont les points de données les plus proches d'un hyperplan.



**Figure 2** : Les vecteurs de support

# HYPERPLAN , VECTEURS DE SUPPORT ET MARGE DANS L'ALGORITHME SVM

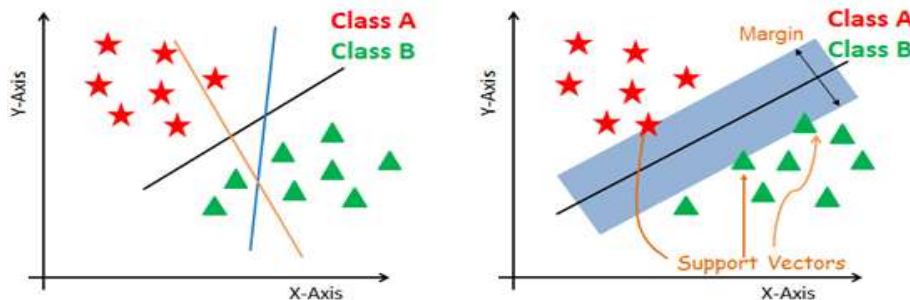
- **Marge** : C'est l'écart entre les vecteurs supports et l'hyperplan. Il peut y avoir plusieurs hyperplans, mais l'objectif de l'algorithme SVM est de sélectionner celui avec la marge maximale car la marge et la précision sont directement proportionnelles l'une à l'autre.



**Figure 4** : Marge dans l'algorithme SVM

# SVM LINEAIRE

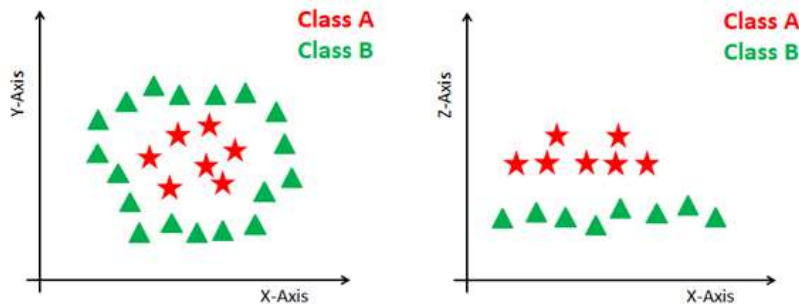
- **SVM linéaire** est utilisé pour les données séparables linéairement, ce qui signifie que les données peuvent être classées en deux classes à l'aide d'une seule ligne droite.
- L' SVM linéaire recherche l'hyperplan marginal maximal dans les étapes suivantes :
  1. Générer des hyperplans qui séparent au mieux les classes.
  2. Sélectionner l'hyperplan droit avec la ségrégation maximale des points de données les plus proches, comme indiqué dans la figure de droite.



**Figure 4:** Fonctionnement de l'algorithme SVM.

# SVM NON LINEAIRE

- **SVM non linéaire** est utilisé pour les données séparées de manière non linéaire, ce qui signifie que les données ne peuvent pas être classées à l'aide d'une ligne droite.
- l'algorithme SVM non linéaire a une technique appelée **l'astuce du noyau**.
- Le noyau SVM est une fonction qui prend un espace d'entrée de faible dimension et le transforme en un espace de dimension supérieure, c'est-à-dire qu'il convertit un problème non séparable en problème séparable en lui ajoutant plus de dimension.

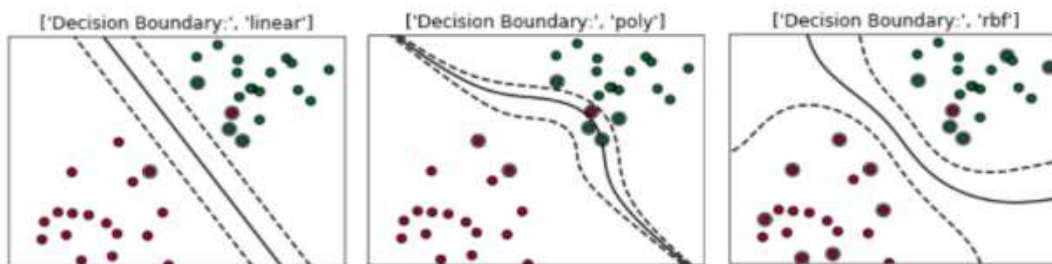


**Figure 5 :** Fonctionnement de l'algorithme SVM : cas non linéaire



# NOYAUX SVM

- Noyau linéaire
- Noyau polynomial
- Noyau sigmoïde
- Noyau de base radiale gaussienne (RBF)
- Noyau d'Anova



**Figure 6 :** Noyaux SVM

# PARAMÈTRES DE RÉGLAGE UTILISÉS AVEC UN NOYAU DANS SVM:

- Il existe deux autres paramètres de réglage utilisés avec le noyau dans un algorithme SVM : la régularisation  $C$  et le gamma.
- **Régularisation  $C$**  : le paramètre de pénalité, qui représente une mauvaise classification ou un terme d'erreur. Le terme de mauvaise classification ou d'erreur indique à l'optimisation SVM combien d'erreur est supportable. Une valeur plus élevée conduit à une classification erronée plus petite, résultant en un hyperplan à faible marge. Au contraire, une valeur plus petite conduit à une classification erronée plus élevée et à un hyperplan à marge élevée.
- **Gamma** : des valeurs plus élevées permettent au SVM de ne considérer que les vecteurs de support proches pour créer un hyperplan, et des valeurs plus faibles prennent en compte même les vecteurs éloignés.

# AVANTAGES / INCONVÉNIENTS DE SVM

## ■ Avantages de SVM :

1. Sa grande précision de prédiction
2. Fonctionne bien sur de plus petits data sets
3. Fonctionne bien avec même des données non structurées et semi-structurées

## ■ Inconvénients de SVM :

1. Sensible au bruit
2. Ne convient pas à des jeux de données plus volumineux
3. L'extension de la classification à plus de deux classes est problématique

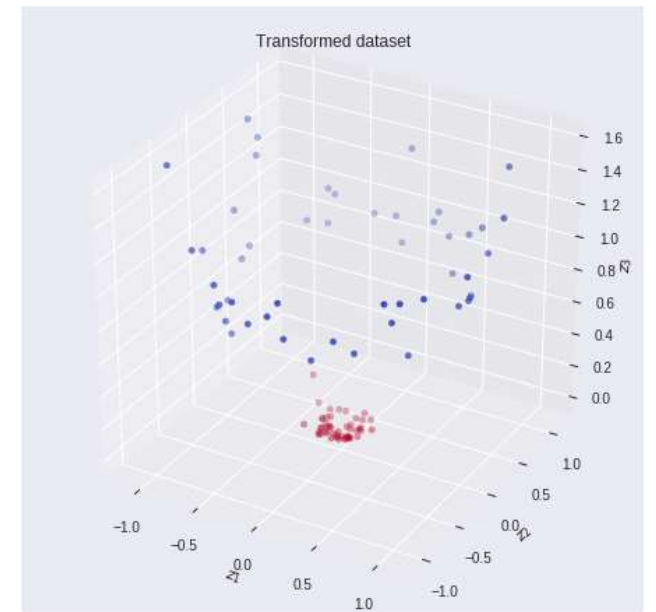
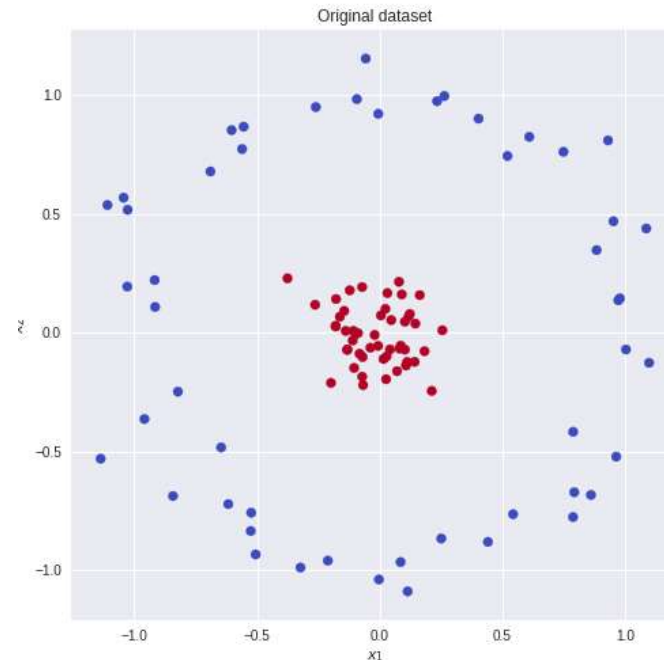
# ANALYSE EN COMPOSANTES PRINCIPALES (ACP): RAPPEL

- L'**ACP** est une méthode d'apprentissage automatique non supervisé bien connue de réduction de dimension qui permet de transformer des variables très corrélées en nouvelles variables décorrélées les unes des autres tout en conservant autant d'informations que possible.
- **Le principe de l'ACP est simple** : Il s'agit en fait de résumer l'information qui est contenue dans une large base de données en un certain nombre de variables synthétiques appelées : Composantes principales. L'idée est ensuite de pouvoir projeter ces données sur l'hyperplan le plus proche afin d'avoir une représentation simple de nos données.
- L'ACP est une méthode **linéaire**. Cela fonctionne très bien pour les ensembles de données séparables linéairement. Cependant, si l'ensemble de données a des relations **non linéaires**, il produit des résultats indésirables.

## ANALYSE EN COMPOSANTES PRINCIPALES DU NOYAU: KERNEL PCA

**Figure 7** : projection des données d'une dimension inférieure (2D) à une dimension supérieure (3D)

- **Kernel PCA** est une variante de l'analyse en composantes principales dans laquelle nous utilisons des méthodes de "**noyau**" pour effectuer une analyse en composantes principales avec des ensembles de données qui ne sont pas linéairement séparables.
- **L'idée de KPCA** repose sur l'intuition que de nombreux ensembles de données, qui ne sont pas linéairement séparables dans leur espace, peuvent être rendus linéairement séparables en les projetant dans un espace de dimension supérieure.



# ÉTAPES DE KPCA :

Soit  $T$  la transformation d'un point vers une dimension supérieure.

■ Les étapes de KPCA :

1. Choisir une fonction noyau  $k(x_i, x_j)$
2. Retrouver la matrice de covariance de données : nous utiliserons la fonction noyau pour calculer cette matrice. Ainsi calculera la matrice du noyau, qui est la matrice qui résulte de l'application de la fonction du noyau à toutes les paires de données.  $K = T(X)T(X)^T$

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_d) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_d) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_d, x_1) & k(x_d, x_2) & \dots & k(x_d, x_d) \end{pmatrix}$$

3. Centrer notre matrice kernel (ceci équivaut à soustraire la moyenne des données transformées et à diviser par les écarts types) :  $K_{\text{new}} = K - 2(I)K + (I)K(I)$ , où  $I$  est une matrice dont tous les éléments sont égaux à  $1/d$ .
4. Trouver les vecteurs propres et les valeurs propres de cette matrice.
5. Trier nos vecteurs propres en fonction de leurs valeurs propres correspondantes dans un ordre décroissant.
6. Choisir le nombre de dimensions que nous voulons que notre ensemble de données réduit soit, appelons-le  $m$ .
7. Choisir les premiers  $m$  vecteurs propres et les concaténerons dans une matrice.
8. Calculer le produit de cette matrice avec les données. Le résultat sera notre nouvel ensemble de données réduit.

# LES KERNEL PCA LES PLUS UTILISÉS

- Linéaire - équivalent à une PCA classique
- Polynomial
- Gaussien (RBF) - le noyau gaussien RBF représente le plus usité
- Laplacien

# RÉFÉRENCES

- <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>
- <https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>
- <https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/>
- <https://iq.opengenus.org/kernal-principal-component-analysis/>
- <https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises/4379521-utilisez-une-acp-avec-un-noyau>
- <https://thecleverprogrammer.com/2021/05/17/kernel-pca-in-machine-learning/>
- <https://www.aionlinecourse.com/tutorial/machine-learning/kernel-pca-in-python>
- <https://analyticsinsights.io/les-svm-support-vector-machine/>